a texture cache memory having addresses partitioned into N banks, each bank containing texels transferred from said main memory that have the corresponding identifier;

a texture cache controller for determining and requesting the necessary transfer of texels from said texture main memory addresses to said texture cache memory addresses, said cache controller including a plurality of least recently used controllers coupled in succession to thereby transfer texels according to a least recently used replacement algorithm; and

a texture cache arbiter for scheduling and controlling the actual transfer of texels from said texture main memory into the texture cache memory and controlling the outputting of texels for each pixel to a interpolating filter from the cache memory.--

All Copt.

REMARKS

This Amendment is filed in reply to the outstanding Official Action of February 22, 2002 and it is believed to be fully responsive to the Official Action for reasons set forth herein below in greater detail.

In the Official Action, the Examiner stated that the references cited in the specification of the above-identified application must be submitted in an Information Disclosure Statement (i.e., "IDS") to be considered by the Office. The Examiner first substantively rejected Claims 1, 2, 5-10 and 13 pursuant 35 U.S.C. §103(a), as unpatentable over Wang, *et al.* (U.S. Patent No. 5,831,640) (hereinafter "Wang") in view Schilling, *et al.* (U.S. Patent No. 6,236,405) (hereinafter "Schilling"). The Examiner further rejected Claims 3 and 4 pursuant 35 U.S.C. §103(a), as unpatentable over Wang in view of Schilling as applied to Claims 1-2, 5-10 and 13 and further in view Thayer, *et al.* (U.S. Patent No. 5,493,644) (hereinafter "Thayer"). The Examiner indicated that Claims 11, 12 and 14-17 would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Finally, the Examiner indicated that Claim 18 is allowed over the prior art of record.

With respect to the references cited in specification of the above-identified application, Applicants are submitting herewith the references in an IDS for consideration by the Office.

At the outset and before addressing particular issues raised in the present Official Action, Applicants have cancelled independent Claim 1 and rewritten allowable Claim 14 in independent form comprising all limitations of Claim 1. Additionally, Applicants have added a new Claim 19, which incorporates the allowable subject matter of Claim 11. More particularly, the amended independent Claim 14 now recites a cache arbiter which is coupled between the cache controller and the cache memory for determining which texels in the cache memory can be overwritten when new texels are determined to be transferred to the cache memory by the cache controller. Support for the amendment is found in the specification on page 17 with reference to Figure 2. Additionally, the new independent Claim 19 now recites a cache controller which includes a plurality of least recently used controllers coupled in succession to thereby transfer texels according to a least recently used replacement algorithm. Support for this amendment is found in the specification on pages 7 and 17. Furthermore, Applicants amended the dependency of Claim 2, thereby making Claims 2-6 (and Claim 15) depend, whether directly or indirectly, from the amended independent Claim 14. Applicants have also amended dependencies of Claims 7 and 8, thereby making Claims 7-10, 12-13, and 16-17 dependent, whether directly or indirectly, from the new claim 19. Applicants respectfully submit that the amendments to the claims add no new subject matter to the above-identified application.

For the foregoing reasons Applicant respectfully requests the Examiner to withdraw the objection to the amended independent Claim 14. Furthermore, Applicant respectfully requests the Examiner to withdraw the 35 U.S.C. §103(a) rejections of claims which respectively, whether directly or indirectly, depend from independent Claims 14, i.e., Claims 2-6 and 15.

Applicants have amended the specification to correct inadvertent misspellings and omissions in the specification. Applicants respectfully submit that the amendments add no new matter to the above-identified application.

Attached hereto is a marked-up version of the changes made to the specification and the claims by the present amendment. The attached marked-up version is captioned: "**VERSION WITH MARKINGS TO SHOW CHANGES MADE**".

In sum, Applicants respectfully submit that the prior art references to Wang, Schilling and Thayer, whether alone or in combination, do not teach or suggest the now amended Claim 14 and the new Claim 19. In view of the foregoing, Applicants believes that this application is in condition for allowance and Applicants henceforth respectfully solicits such allowance. If the Examiner believes a telephone conference might expedite the allowance of this case, Applicants respectfully requests the Examiner to call the undersigned, Applicants' attorney, at the following telephone number: (516) 742-4343.

Respectfully submitted,

Paul J. Esatto, Jr.
Registration No. 30,749

Scully, Scott, Murphy & Presser
400 Garden City Plaza
Garden City, New York 11530
(516) 742-4343

AGV:gc

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION:

The paragraph beginning on page 4, line 4, has been amended as follows:

--The problem with texture mapping systems that accomplish the texture addressing and look-up in a brute force method, is that the transfer of data between the various processes require wide memory buses, multiple and/or multi-ported memory subsystems, and/or multiple [clocks] clock cycles to transfer the required data. Some of these systems also are penalized with the time required to load the texture maps or large chunks of the maps into a specialized memory system prior to rendering of primitives that use it. Texture mapping has been accomplished with special purpose memory devices or a special purpose memory system that can offer effective random access anywhere within a texture map.--

The paragraph beginning on page 16, line 6, has been amended as follows:

--Referring now to Figure 2, there is shown a block diagram of the three-dimensional texture caching system of the present invention. The cache system 24 includes a cache controller 26, a cache arbiter 28 and a texture cache memory 30. Also shown in Figure 2 [is] are a texture address calculator 32, a texture main memory 34 and an interpolator 36. The texture address calculator 32 calculates the texture memory addresses (U,V) for each pixel and also the specific LODs from which the texture addresses are to be retrieved. The cache controller 26 determines if the cache memory 30 will contain the texture data for a given address or if it needs to be fetched. It allocates

space for new data to be fetched and determines the location of the data required for each pixel to pass tag data to the cache arbiter 28, which will be used to schedule data entry and access from the cache memory 30. The cache controller 26 also determines the order in which data will be overwritten. Based on this determination, the cache controller 26 fetches data from the texture main memory 34 and transfers it into the appropriate addresses in the texture cache memory 30 when that cache arbiter 28 determines the data located in that location is no longer in use. In accordance with a preferred embodiment of the present invention, the cache controller 26 is comprised of four individual controllers 38, 40, 42 and 44, each of which [correspond] corresponds to the partitioning of data into four different data groups in the texture main memory 34. This partitioning of data also applies to the four memory banks in the texture cache memory 30. The controllers 38,40,42,44 regulate and keep track of what is stored in the memory banks of the cache memory 30. In the preferred embodiment, each of the controllers contains four stages with each stage referencing a double quad word in the respective cache memory bank. The memory banks are organized into an array of W, X, Y and Z texel blocks, partitioned by A, B, C, and D texels in each array. The cache arbiter 28 determines if it can write the next data values into the cache without overwriting any data that is still needed. Once the appropriate texture data is determined to be present in the cache memory 30, the controller outputs the appropriate texel data into the interpolator 36 to perform the required interpolation such as bilinear interpolation.--

The paragraph beginning on page 17, line 23, has been amended as follows:

--Figure 3 shows a texture map that is to be stored in main memory 46 being partitioned by labeling or identifying each texel with four different identifiers. As shown, texels are labeled A, B, A, B etc. across every even row of the map and likewise C, D, C, D etc. across every odd row of the map. This organization results in an interesting relationship when bilinear interpolation is performed. After mapping a pixel into the texture map, the texture addresses to the texels that surround the mapped pixel result in a group of texels, no matter what four are selected, that will always be consisting on one A, one B, one C, one D type of texel. This allows the organization of texel types in cache memory to be in banks so that all four texels that are needed for a pixel can be accessed in one clock cycle. This is accomplished by putting all A types in one bank, all B types in another bank, C types in a third bank, and D types in a fourth bank. The mapping of pixel 48 into texture map 46 is an example showing that one of each texel type will be selected for bilinear interpolation.--

The paragraph beginning on page 19, line 19, has been amended as follows:

--In accordance with the present invention, an optimal texture block size can be determined to simplify caching. From a topological point of view and from the random nature of the alignment of texels and pixels, the following relationship should be satisfied in order to maintain good texture coherency. First, the texel block arrangement should be compact such as a square or at most a two to one aspect ratio rectangle. Second, the texel block size should be compatible or multiples with the texture data transfer size per clock. Third, the texture main memory array should be organized so that

it can be retrieved one block at a time from a single page of memory to allow the use of inexpensive memory devices. For example, if the [buss] bus width for transfer of data from main memory to cache memory is 128 bits wide per clock cycle, the number of transfers per request and the texel depth (size in bits per single texel) determines the organization. If texels are 8, 16 or 32 bits and the number of transfers per request is 1 then the optimal block sizes are 4x4, 2x4, and 2x2 respectively. The texture main memory array can be made from inexpensive memory devices because the data is organized for continuous block transfer per clock cycle.--

The paragraph beginning on page 21, line 18, has been amended as follows:

--As shown in Figure 5, when texture data is moved from texture main memory to texture cache memory, the texels 50 undergo a conversion or reorganization. They are stored in the texture main memory array so that a double quad word contained in texel block 52 can be accessed and sent across the [buss] bus 64. The texture cache memory storage requires a reorganization of the elements so that they are stored in a form necessary for access per clock cycle in the bilinear texture interpolator. This is achieved by storing the texels of the double quad word in four cache memory banks 56,58,60,62. A double quad word is made up of an even and an odd part. The A texels of double quad word 52 are stored in bank A, the B texels are stored in bank B, the C texels are stored in bank C and the D texels are stored in bank D.--

The paragraph beginning on page 31, line 4, has been amended as follows:

--Since it can take eight [clocks] clock cycles in a worst case to load all the data necessary for one pixel, a conflict may cause a missed pixel processing clock cycle. This can only happen in very special cases; e.g. starting up a new polygon, when the LOD changes on a polygon or when in a non-LOD mode.--

The paragraph beginning on page 31, line 11, has been amended as follows:

--Through an exhaustive series of simulations it has been shown that very few [clocks] clock cycles are missed and the process seldom requires clock delays. Most of the time the data is already in the cache waiting to be used. This fact is due to the nature of the texture and pixel coherency.

IN THE CLAIMS:

Claims 2, 7, 8 and 14 have been amended as follows:

2. (Amended) The system of claim [1]14, wherein the system further includes a texture addressing scheme for organizing the array of texels in main memory to group spatially related texels in one memory page.

7. (Amended) The system of claim [1]19, wherein N is equal to four and said texture main memory is organized into a plurality of texel blocks each having one of four block texel cache memory identifier in accordance with the following criteria: each texel block consisting of at least one group of four contiguous texels, the texels in each group consisting of one of each of the per texel cache memory identifiers, and wherein said texture cache memory being partitioned into a plurality of rows corresponding to said plurality of block texel cache memory identifiers, each cache memory bank having at least one row corresponding to each of the four block texel cache memory identifiers.

8. (Amended) The system of claim [1]19, wherein said cache controller includes N stages.

14. (Amended) [The] A computer graphics processor system [of claim 1,] having the capability of mapping texture onto a three dimensional object in a scene being displayed, the system comprising:

a texture address calculator for generating texel addresses for a list of primitives being processed;

a texture main memory containing an array of texels, each texel having an address and one of N identifiers;

a texture cache memory having addresses partitioned into N banks, each bank containing texels transferred from said main memory that have the corresponding identifier;

a texture cache controller for determining and requesting the necessary transfer of texels from said texture main memory addresses to said texture cache memory addresses; and

a texture cache arbiter for scheduling and controlling the actual transfer of texels from said texture main memory into the texture cache memory and controlling the outputting of texels for each pixel to a interpolating filter from the cache memory, said [wherein the] cache [read/write] arbiter [is] coupled between said controller and said texture cache memory for determining which texels in the cache memory can be overwritten when new texels are determined to be transferred to said cache memory by said cache controller.